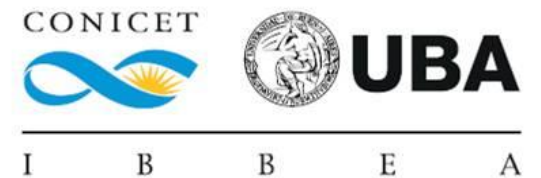
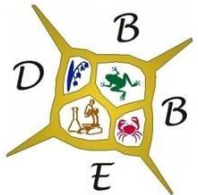


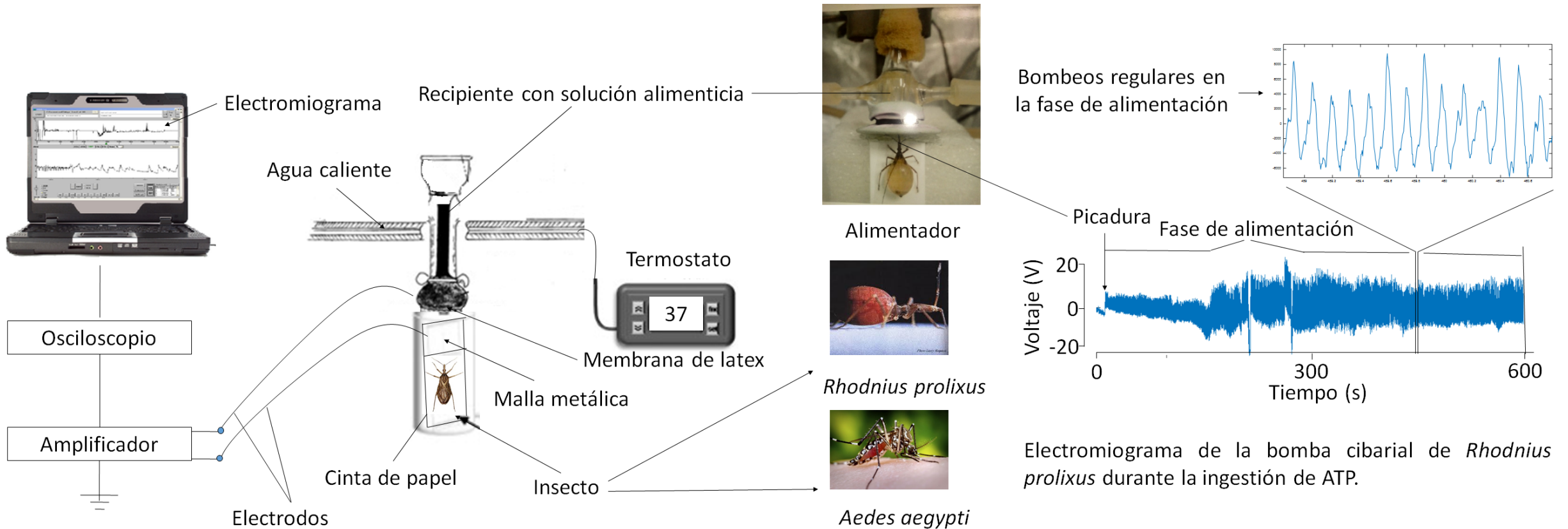


Facultad de Ciencias Exactas y  
Naturales

# Redes Neuronales Convolucionales y Recurrentes para el Análisis de Electromiogramas del Aparato Bucal de Vinchucas



# Sistema para el registro de datos

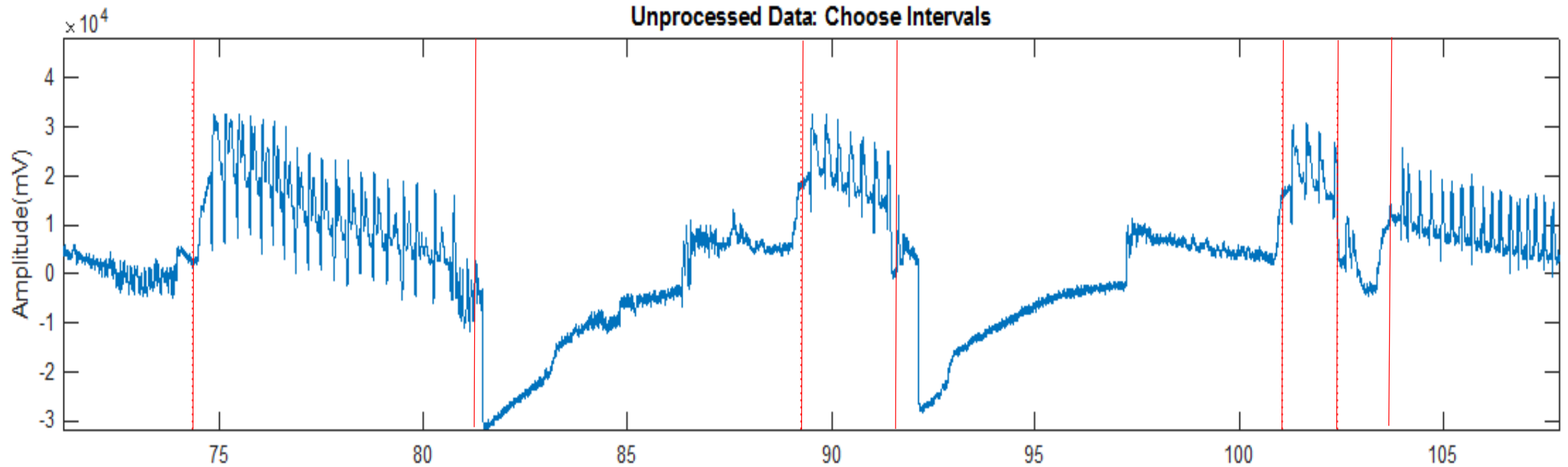


# **Variables que pueden medirse a partir de los Electromiogramas**

- a. Número de veces que el insecto pica
- b. Tiempo de cada evento de picado
- c. Tiempo de bombeo
- d. Tiempo de no-picado
- e. Número de bombeos totales
- f. Frecuencia de bombeo (durante c)
- g. Amplitud media de bombeo

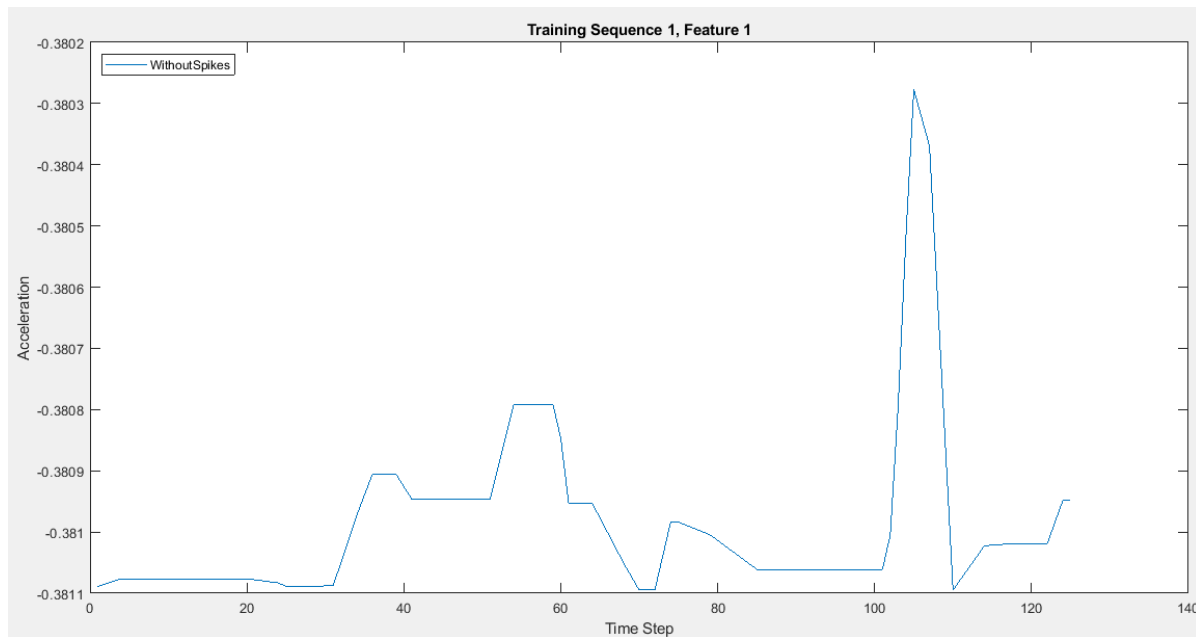
# Definir intervalos temporales de interés

Los intervalos temporales donde se miden estas variables se definen manualmente.

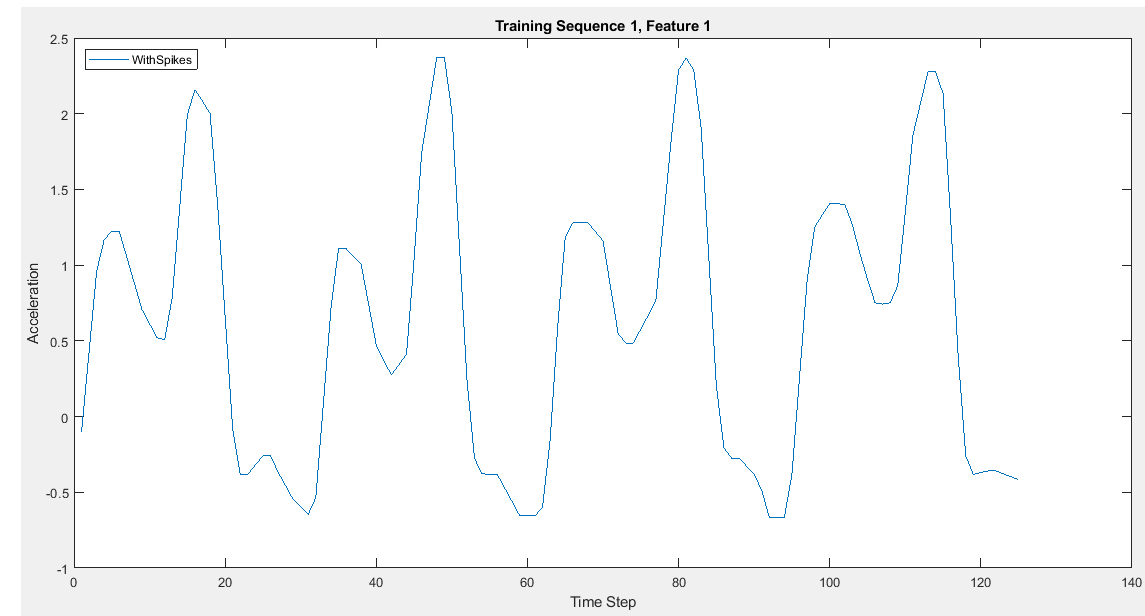


# El objetivo consiste en elaborar un modelo de red neuronal capaz de reconocer dos clases:

## Ausencia de Picos



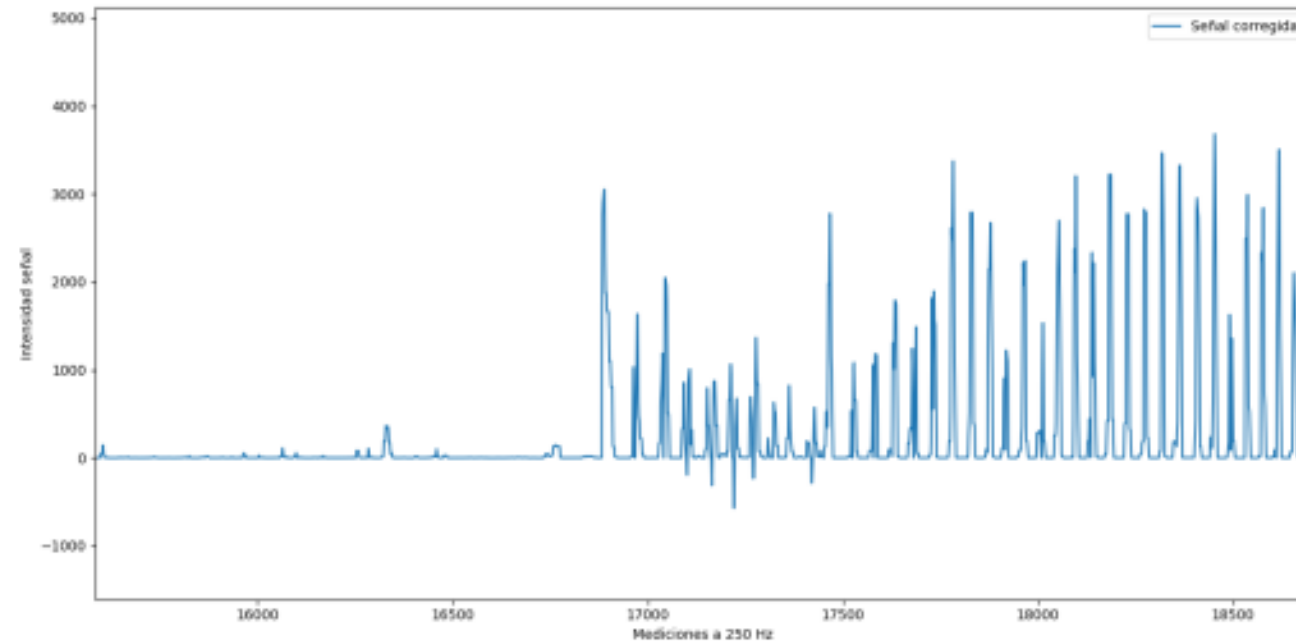
## Presencia de Picos



## El problema de clasificación

Automatizar el proceso de detección inicio/fin de región de interés (con picos).

Distinguir regiones sin pico de regiones con pico.

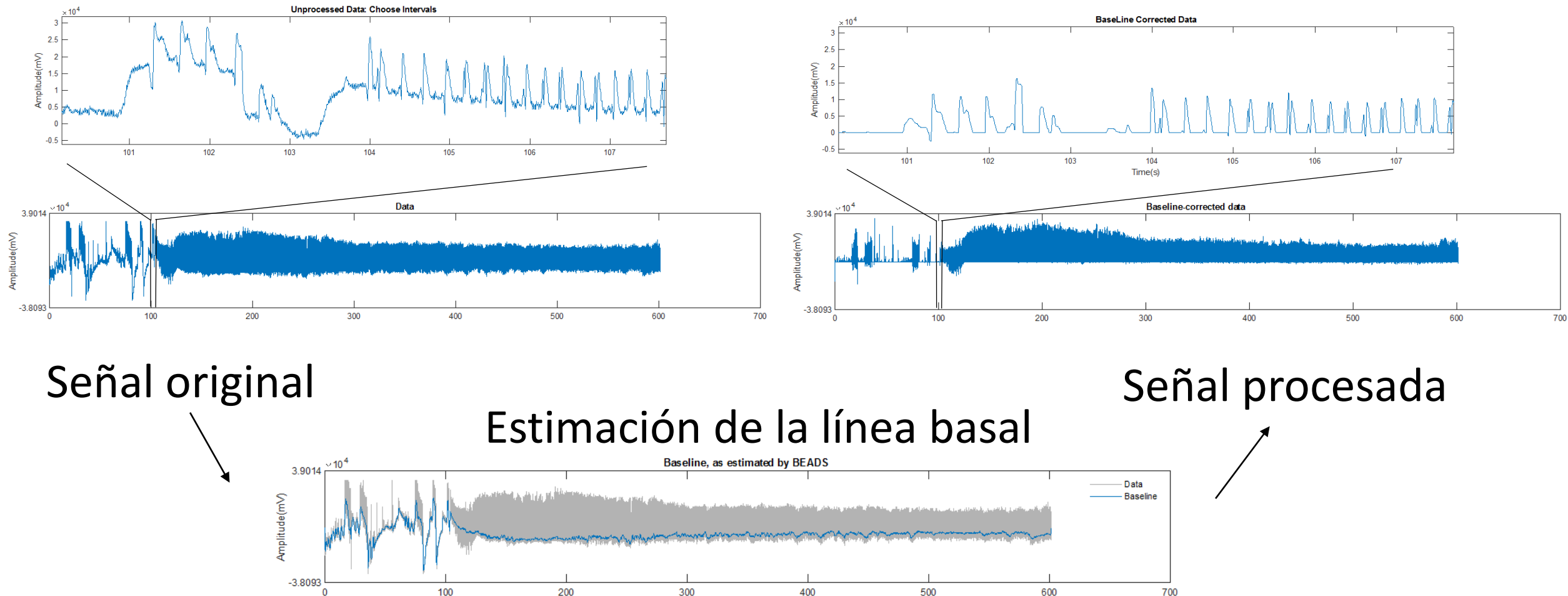


Sin pico: el insecto no está comiendo.

Con pico: el insecto está comiendo (región de interés).

# Preprocesamiento de la señal

Eliminar ruido, estimar la línea basal de la señal y corregir su deriva mediante el algoritmo BEADS (Baseline Estimation and Denoising with Sparsity).

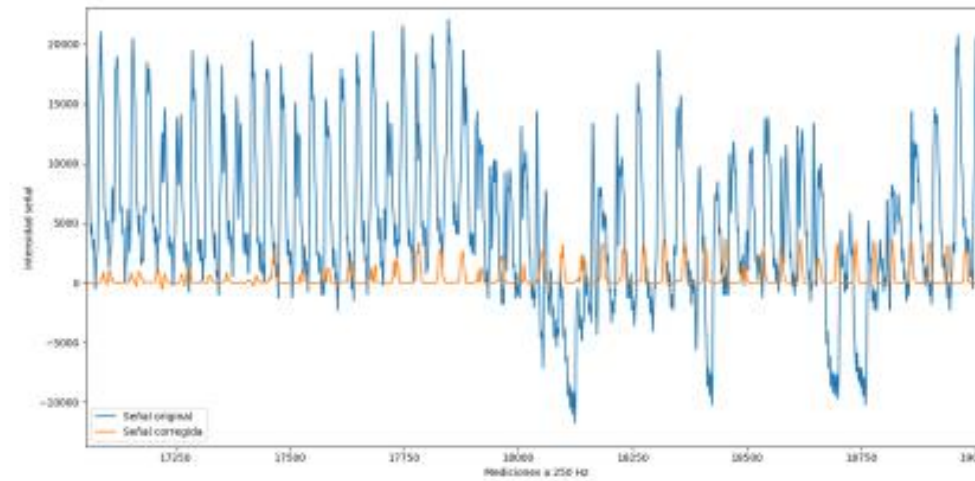
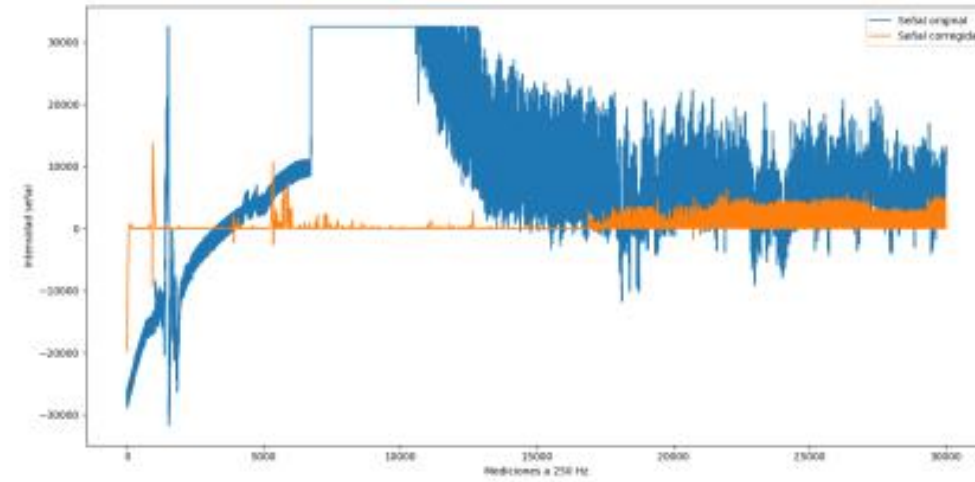


Señal original

Estimación de la línea basal

Señal procesada

# Preproceso de las series por BEADS<sup>1</sup>



---

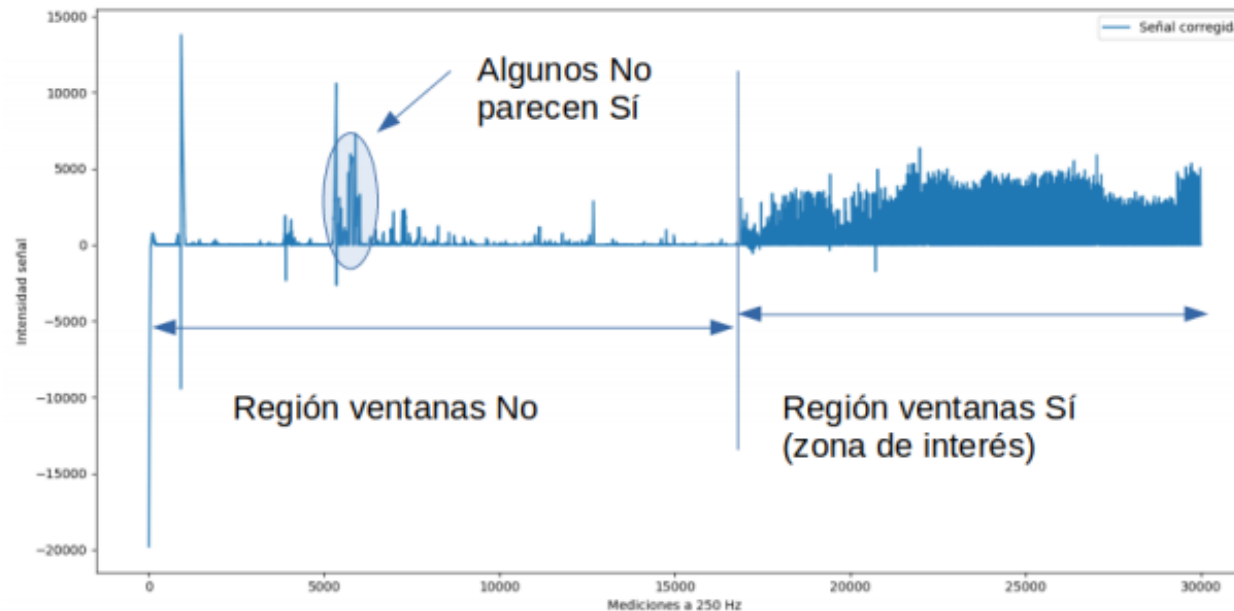
<sup>1</sup><http://www.laurent-duval.eu/siva-beads-baseline-background-removal-filtering-sparsity.html>



# División de la señal en ventanas temporales de 0.5 y 1 segundo de duración

Partición en ventanas para clasificar

Se parte la serie en ventanas, cada una clasificada Sí/No región de interés.



Luego a la primera Sí: inicio región de interés  
A la primera No: fin región de interés

# Modelos de Redes Neuronales Construidos

## Primeras redes

Redes convolucionales con estas arquitecturas:

- ▶ Input: ventanas de 6 seg. O sea serial temporal 1D de 1500 datos (250 Hz).
- ▶ Convolutacional 1D de 5x60 (stride=1 o 2, padding=20)
- ▶ ReLu
- ▶ Convolutacional 1D de 10x30 (stride=1, padding=10)
- ▶ Max Pooling de 4 o de 2
- ▶ ReLu
- ▶ FC de 1830 a 300 (luego tanh)
- ▶ Prueba agregando FC a 300/400
- ▶ FC de 300 a Sí/No

Accuracy de aprox. 90-92 %.

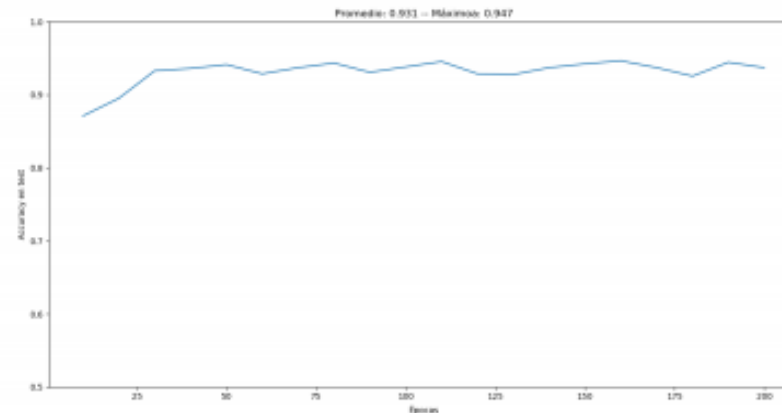
# Modelos de Redes Neuronales Construidos

## Segundas redes: algo más profundas

Una red convolucional con esta arquitectura:

- ▶ Input: ventanas de 6 seg. O sea serial temporal 1D de 1500 datos (250 Hz).
- ▶ Convolutacional 1D de 5x60 (stride=1, padding=30)
- ▶ ReLu
- ▶ Convolutacional 1D de 10x30 (stride=2, padding=20)
- ▶ Max Pooling de 2
- ▶ ReLu
- ▶ Convolutacional 1D de 20x15 (stride=2, padding=30)
- ▶ Max Pooling de 2
- ▶ ReLu
- ▶ FC de 580 a 300 (luego tanh)
- ▶ FC de 300 a 100 (luego tanh)
- ▶ FC de 100 a Sí/No

Accuracy de  
aprox.  
93-94 %.



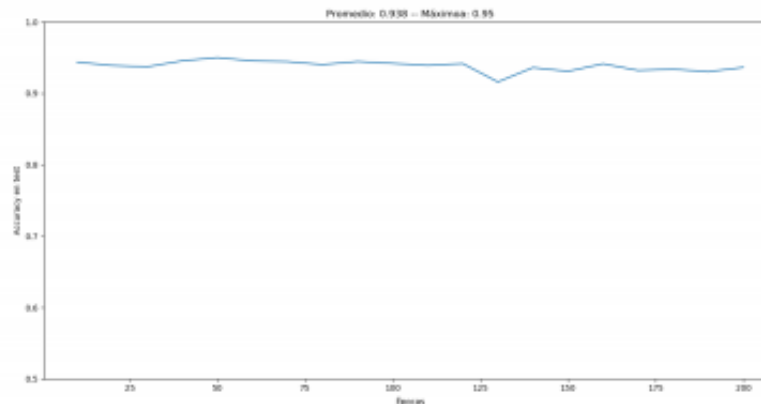
# Modelos de Redes Neuronales Construidos

## Terceras redes: con ventanas de 1 seg.

Una red convolucional con esta arquitectura:

- ▶ Input: ventanas de 1 seg. O sea serial temporal 1D de 250 datos (250 Hz).
- ▶ Convolucional 1D de 5x60 (stride=1, padding=30)
- ▶ ReLu
- ▶ Convolucional 1D de 10x30 (stride=2, padding=30)
- ▶ Max Pooling de 2
- ▶ ReLu
- ▶ Convolucional 1D de 20x15 (stride=2, padding=30)
- ▶ Max Pooling de 2
- ▶ ReLu
- ▶ FC de 580 a 300 (luego tanh)
- ▶ FC de 300 a 100 (luego tanh)
- ▶ FC de 100 a Sí/No

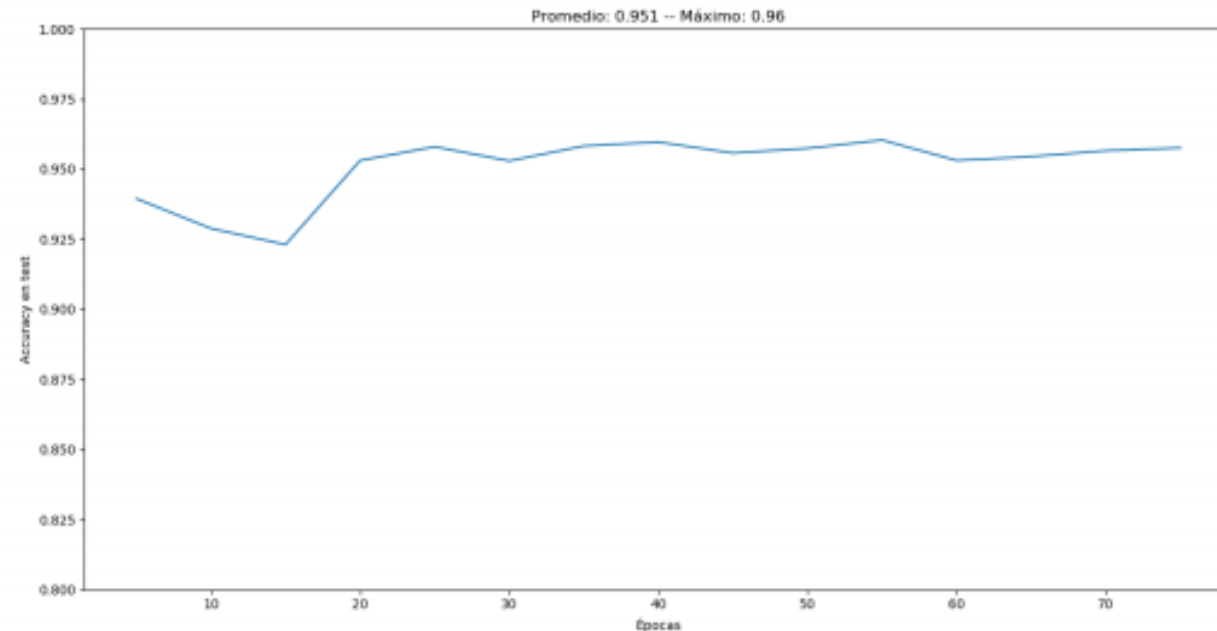
Accuracy de  
aprox. 94 %.



# Modelos de Redes Neuronales Construidos

Terceras redes: con ventanas de 1 seg.

Ídem antes, ahora normalizando los datos  
Accuracy de aprox. 95-96 %.



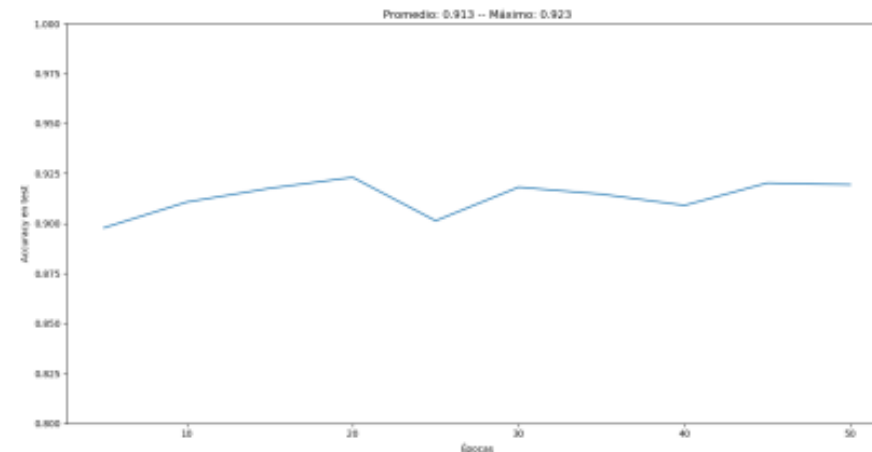
# Modelos de Redes Neuronales Construidos

Cuarta red: con las FFTs de las ventanas de 1 seg.

Prueba rápida anoche:

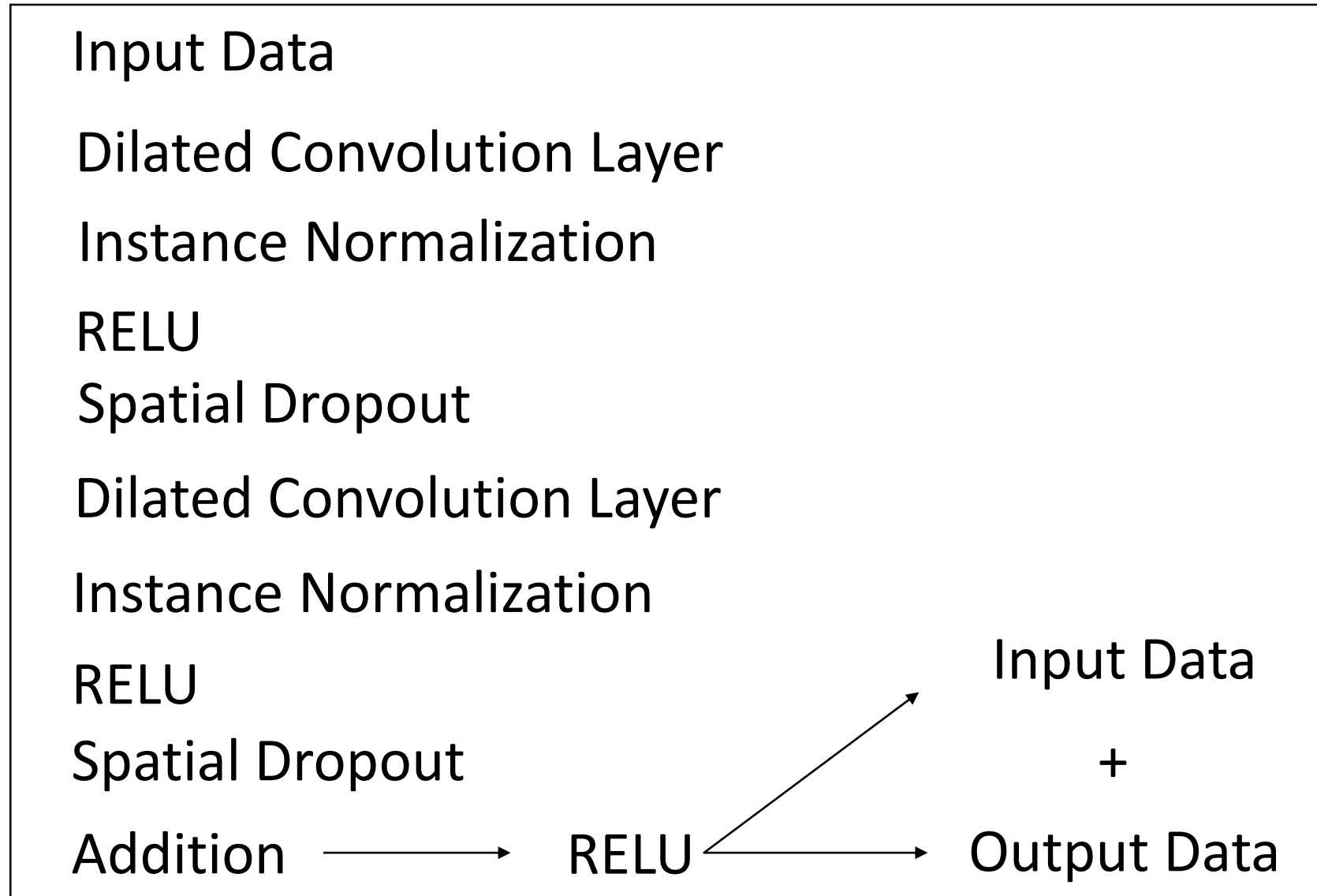
- ▶ Input: FFTs de las ventanas de 1 seg.
- ▶ Convolutacional 1D de 5x60 (stride=1, padding=30)
- ▶ ReLu
- ▶ Convolutacional 1D de 10x30 (stride=2, padding=30)
- ▶ Max Pooling de 2
- ▶ ReLu
- ▶ Convolutacional 1D de 20x15 (stride=2, padding=30)
- ▶ Max Pooling de 2
- ▶ ReLu
- ▶ FC de 580 a 100 (luego tanh)
- ▶ FC de 100 a Sí/No

Accuracy de aprox. 91-92 %.



## **Otro Modelo de Red Convolutiva**

## Stack de 4 bloques



Número de filtros:  
175

Tamaño del filtro:  
3

Stride: 1

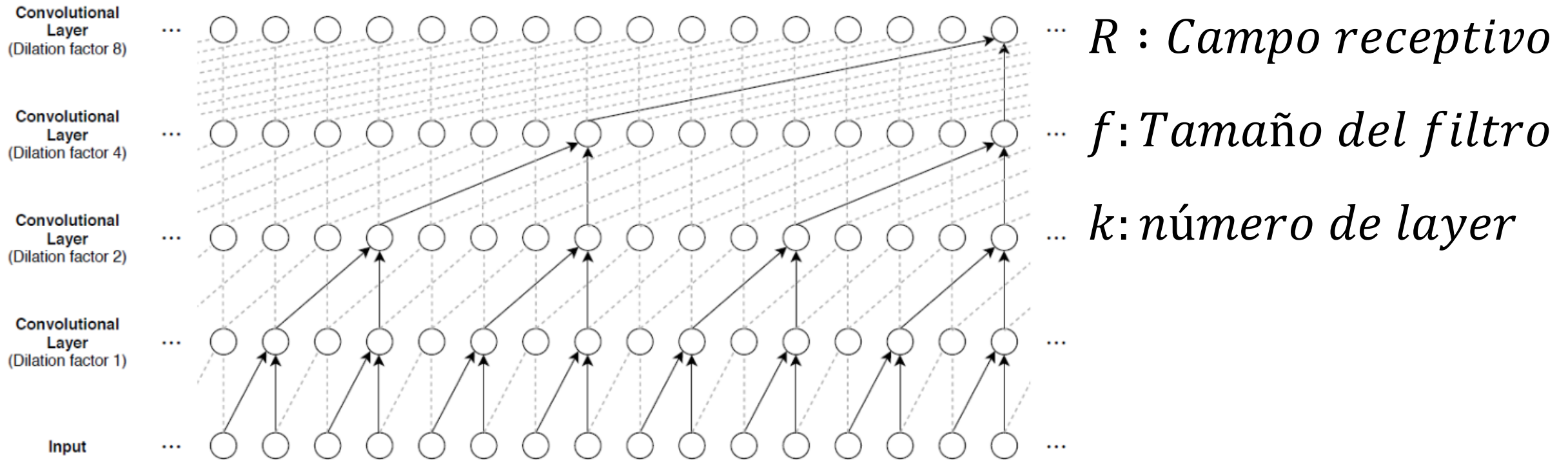
Classes: 2

Fully Connected Layer



# Otro Modelo de Red Convolucional

Dilated Convolution Layer  $\longrightarrow R = (f - 1)(2^k - 1) + 1$



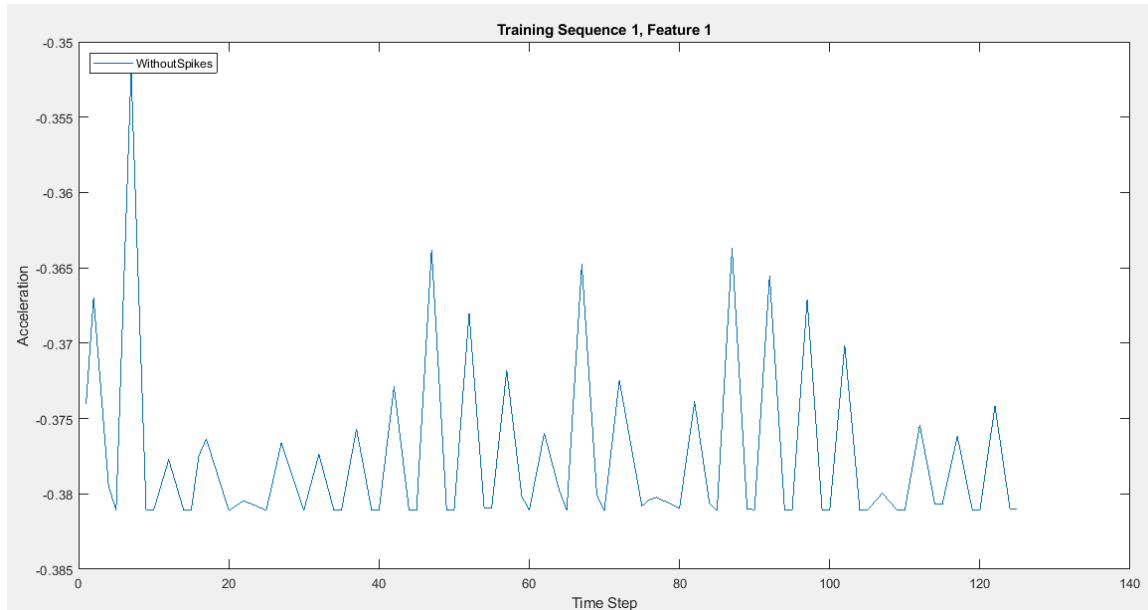
$$R = \text{Padding}$$

# Formato de los datos de entrada en el dominio temporal

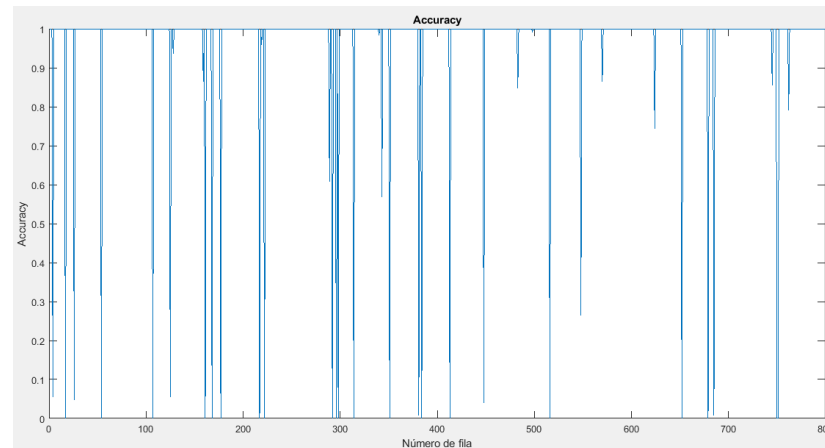
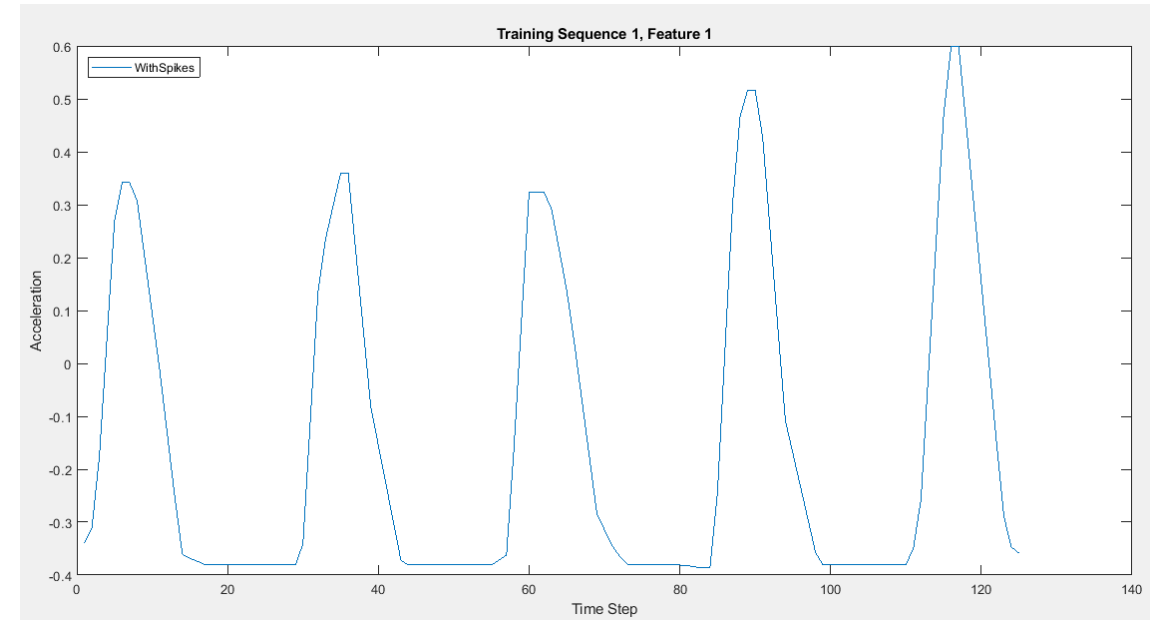
[illegible]

# Resultados

## Ventana sin picos



## Ventana con picos

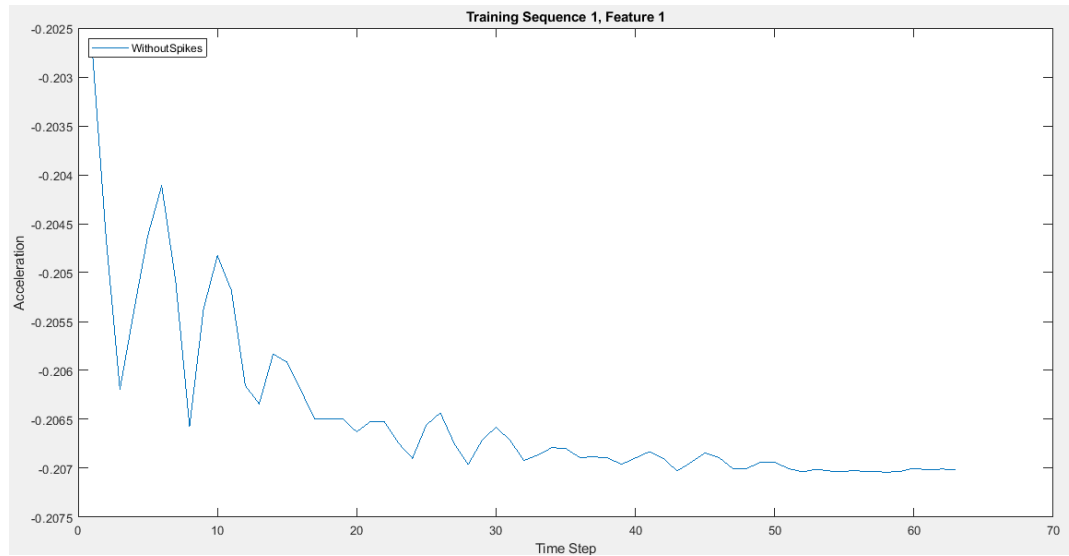


Accuracy: 0.9644

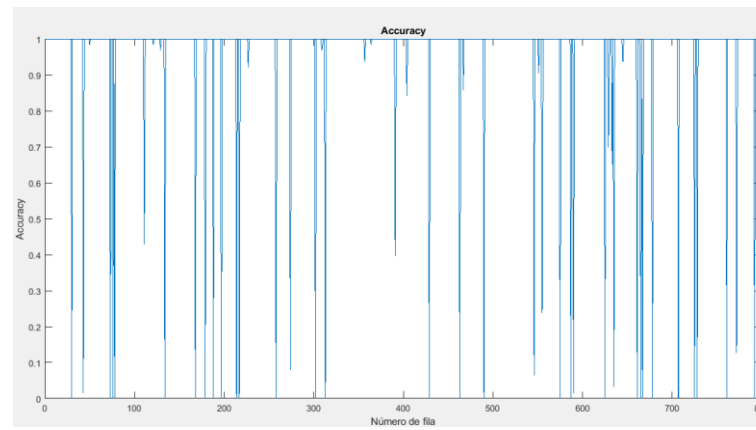
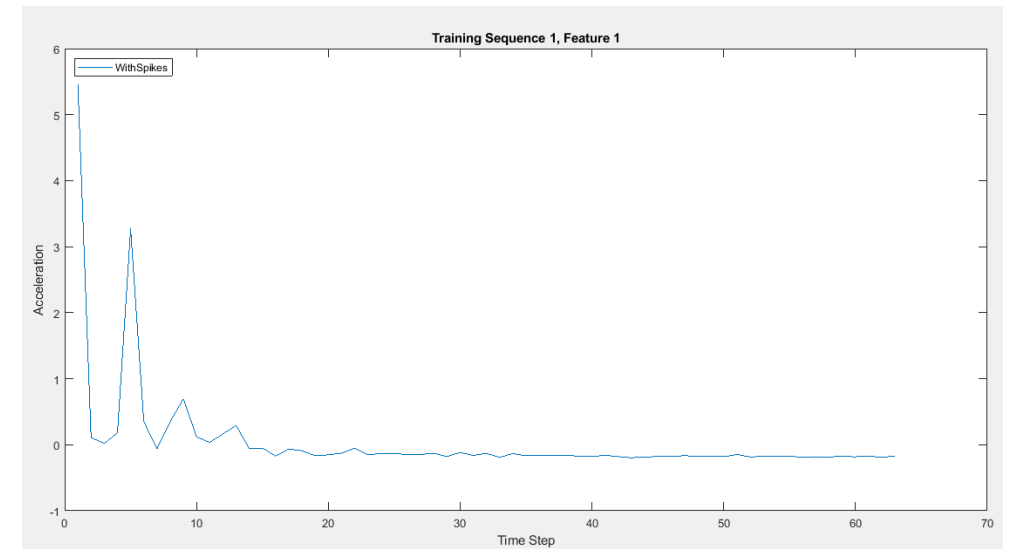
# Resultados

Aplicando la transformada de Fourier a las filas

Ventana con picos



Ventana con picos



Accuracy: 0.9523

# Modelo de Red Recurrente

Input Data

LSTM Layer

Instance Normalization

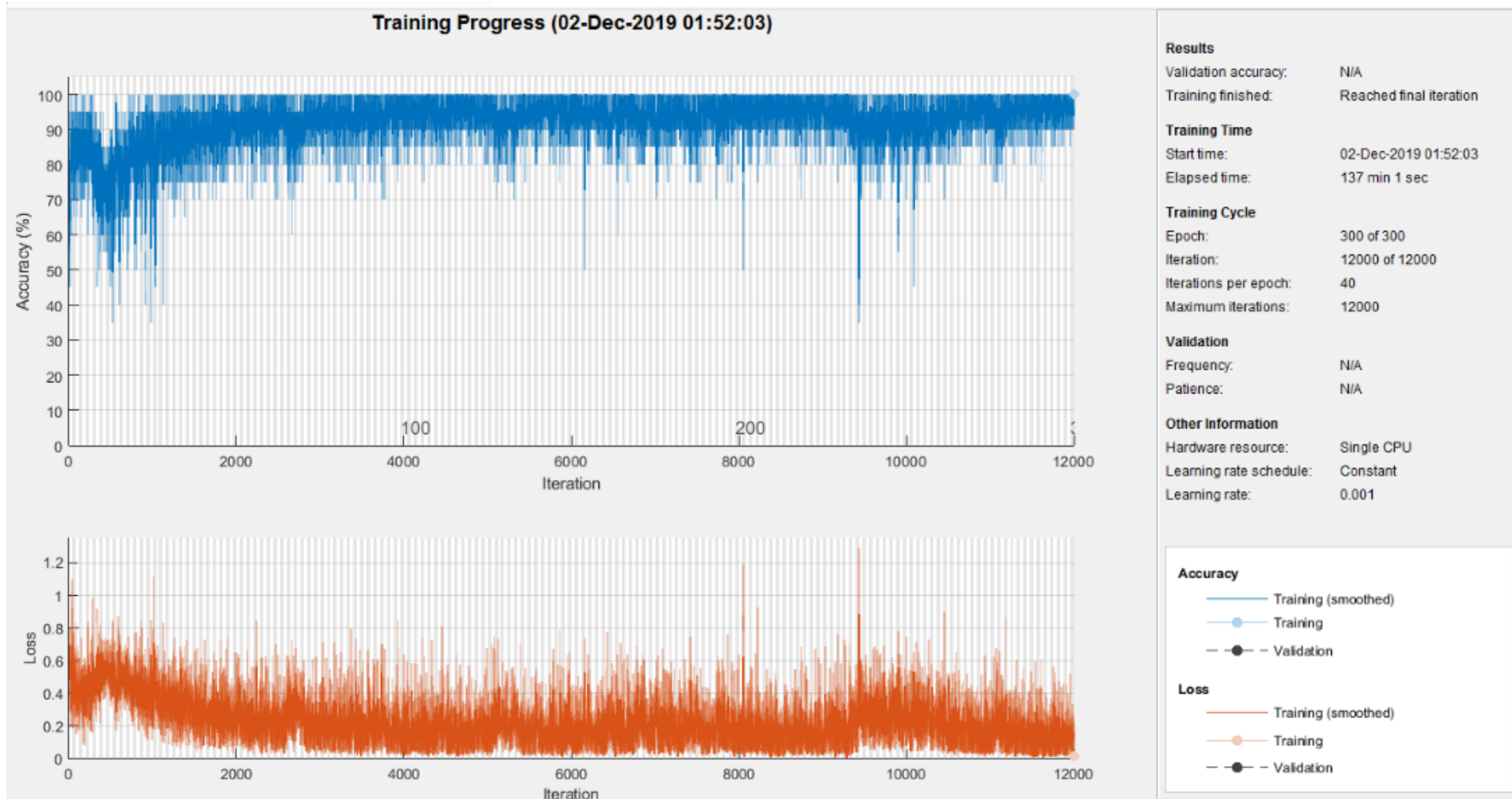
Fully Connected

Softmax Layer

Classification Layer

Número de Unidades Ocultas:  
100

# Resultados



Accuracy: 0.9300

## ***Conclusiones y pasos futuros:***

- ▶ Probar otras Recurrentes.
- ▶ Input con series temporal + FFTs.
- ▶ Ensamblés.
- ▶ Barrer el tamaño de la ventana más exhaustivamente.
- ▶ Barrer tamaño de batch más exhaustivamente.

**¡GRACIAS!**